**Preconditions:**

Node js installation and corresponding environment variable should be set.

**How to run:**

In visual studio code, under terminal tab -> node filename.js

**How to write code:**

Create a new file and save it with extension .js

1. To print something in the console

Console.log(“Hello World!”);

1. To run a js file 🡪 node filename.js
2. To write comments

Add 2 forward slash for one line comments

For multi-line, /\* kasjdhadadk

Sdkajldjasldkjl\*/

1. Variable types – Var (till JS engine 5) and with recent version var, let and const (from JS engine 6) where let and const are more preferred.
2. Let a = 4, let b = “Ragav”, let c = true and console.log(a)
3. Typeof(variable name) // returns you the variable type like number, string or boolean
4. Null and undefined are other two types where undefined is something where user just declared the variable and doesn’t initiate it.
5. Difference between var and let.

When var is used you can redeclare a variable whereas with “let” you can’t redeclare a variable.

Var a= 10; var a= 20; //redeclare and reassigning is possible

Let a=20; let a=30; // invalid JS will throw an error stating that variable a is declared already.

**Var** – scope is at global and function level i.e., when user creates a variable at global level, it can be accessed anywhere and when declared & initialized inside the function, it is accessible only at the function level only. For eg., if you declare and initialize a variable inside the function, then if user tries to print that variable it will throw error as the scope lies within the function.

**Note**: if the same has been declared inside the if block it is not considered as a function hence whatever declared inside the if block, then it will still accessible

**Let –** scope is at global / block level (if declared inside a block then it is not visible outside the block).

Note: Const will not allow either redeclare or reassign as well. It will be used only for constant value.

1. If and while loop

If(true){

Console.log(“inside if”)}

Else{

Console.log(“inside else”)}

Let I = 0;

While(i<10){

I++

console.log(“inside while”)}

1. Do While loop

do{

i++

}while();

Console.log(i)

**Array:**

1. How to declare an Array?

Let marks1 = Array(6)

or

Let marks = new Array(1,2,3,4,5,6)

or

Let marks = [1,2,3,4,5,6] //effective way to declare an array.

Console.log(Marks[2]) //output is 3

1. To assign a new value based on index on the array

Marks[2] = 14

Console.log(marks[2]) //output is 14

1. How to find the length of an array

Console.log(Marks.length) //output is 6

1. How to increase the length of an array and add a new element at the end

Marks.push(65)

Now marks.length ==> 7

1. To print an array

Console.log(marks)

1. To delete the last element from the array

Marks.pop() // will delete the last element

1. To add element at the beginning

Marks.unshift(12) //will add 12 as the first element.

1. How to find the index of a particular element

Marks.indexOf(12) // output is 1 because 12 sits at first index.

1. To check whether a particular element is present in an array.

Marks.includes(12) //output is true.

1. How to get the part of an array

Var marks = [1,2,3,4,5,6]

subMarks = Marks.slice(2,5) //starting index is same but ending index will be ending index – 1.

console.log(subMarks) //3,4,5

1. To iterate through an array using for loop

For(let i=0;i<marks.length;i++){

Console.log(marks[i])}

1. To sum all elements in the array

Var sum=0

For(let i=0;i<marks.length;i++){

Sum = sum+marks[i]

}

Console.log(sum)

1. To sum all elements in the array using reduce method.

Let total = marks.reduce((sum1,mark1) => sum1+mark1,0)

where reduce is a method, “sum1” and “mark1” are two variables in which “sum1” was set to “0” (mentioned at the end) & keep storing the added value for each iteration and “mark1” will keep fetch element from array in each iteration.

1. How to print the even numbers from an array

Var scores = [12,13,14,16]

Var newScores=[]

For(let i=0;i<scores.length;i++){

If(scores[i]%2 ==0){

newScores.push(scores[i])

}}

Console.log(newScores)

1. How to achieve the above with filter method on array

Var scores=[12,13,14,16]

Var newEvenScore = Scores.filter(evenScore => evenScore %2 ==0)

Console.log(newEvenScore)

1. How to use map in array which is used to change all elements into a new value i.e., multiple each element by 3 and update the value.

Let us say now you got the new array in above step which has only even numbers.

Var mapArray = newEvenScore.map(score => score\*3)

Console.log(mapArray)

1. How to use all methods in one line i.e., chaining

Var score = [12,13,14,16]

Var newscorearray = Score.filter(newScore => newScore%2 ==0).map(scoree => scoree\*3).reduce((sum,marks)=>sum+marks,0)

1. How to sort an array of strings

Let fruits = [“banana”,”pear”,”apple”]

Fruits.sort()

Fuits.reverse() // will give the output in descending order.

Console.log(fruits) //[“apple”,”banana”,”pear”]

1. How to sort an integer array

Let marks = [2,6,19,56,2,44]

Console.log(Marks.sort((a,b)=>a-b)) //bubble sort in recursive manner

1. **Functions:**

**Syntax**

function add(a,b){

return a+b

}

Where function is the keyword and add is the name of the method.

Alternate ways

1. Let sumofIntegers = function(a,b){ //anonymous fn ex

{

return a+b

}

1. Let sumofNumbers=(a,b)=>a+b //anonymous fn with fat pipe operator

Console.log(sumOfNumbers(2,3))

**Strings:**

1. How to find the length of the string

Let name = “Ragavendran”

Console.log(Name.length)

1. How to get the substring

Name.slice(0,4) //starting index is same but ending index is index – 1

1. How to get the particular letter from a string based on index value

Console.log(name[1]) //a

1. How to split string based on condition

Let newname = Name.split(‘v’)

Console.log(newname[0]) //Raga

1. How to remove leading and trailing spaces

Console.log(name.trim())

1. How to convert a string into an integer using parseInt method

Let day1 = “23” //in string format

Let day2 = “27” //in string format

Let diff = parseInt(day2) – parseInt(day1) //in number format

Diff.toString() //to convert a number into a string

1. Concatenate

Let day = “Tuesday” + “is funday”

Console.log(day) //Tuesday is funday.

1. Use of indexof

Let day = “Tuesday” + “is funday”

Console.log(day) //Tuesday is funday.

day.indexOf(“day”) // 4 🡪 because it will check the first occurrence of letter “d”

day.indexOf(“day”,5) //14 🡪 it will start from 5th hence it searched in funday

Note: output will be -1 if there is no match has been found

1. Scenario – How will you check the total occurrence of the word day in a string

Let day = “Today is Tuesday and it is a funday”

Let count = 0

Let val = day.indexOf(“day”)

While(day!==-1){

Count++

Val = day.indexof(“day”,val+1)

}

**How to JS create objects:**

let person =

{

Firstname : ‘ragavendran’,

lastname : ‘venkatesan’,

}

How to access the keys:

Console.log(Person.firstname) //Ragavendran

Another way to access

console.log(person[firstname])

**How to update a key value**

person.firstname = “rumble’;

**How to add a new key:**

person.gender = “male”;

**How to print the JS object**

console.log(person); //{firstname:’Rumble’, lastname:’venkatesan’, gender:’male’}

**How to delete a key**

delete person.gender

console.log(person); //{firstname:’Rumble’, lastname:’venkatesan’}

**How to check a property exist in JS object:**

Console.log('gender’ in person) //false

Console.log(‘firstname’ in person) //true

**How to print all the values of a property using for loop:**

For(let key in person){

console.log(person[key]);

}

**Note:** The value can be a string, number or even a function.

**How to assign a function as a value for a key in JS object:**

let person = {

firstname : ‘Ragavendran’,

lastname : ‘venkatesan’,

age: 24

fullname : function(){

console.log(this.firstname + this.lastname) //to access the key inside the object, we need to use this keyword

}}

Console.log(person.fullname()) //remember to put brackets at the end.

**Class in JS:**

Class Person{

age = 38;

location = “Chennai”;

get location(){

return “chennai";

}}

let person = new Person();

person.age;

(person.location) //no need to put the brackets since it’s a property and not a method

**Constructor:**

It is used to assign values and get called when user creates an object.

Class Person{

Constructor(firstname, lastname){

This.firstname = firstname;

This.lastname = lastname:

}

Fullname(){

Console.log(this.firstname + this.lastname);

}

}

Let person = new Person(‘tim’,’joe’);

Person.fullname();

**How to export a class:**

To export a class

Change the first line of the file as

Module.exports = class Person{

Have all the code

}

In another class import it by using

Const AnotherclassPerson = require(‘./filename’) //to import into another file

Let person = new anotherclassPerson(“chris”,”jor”)

Console.log(person.fullName()) //fullname() from class which we exported

**Inheritance:**

Const person = require(“./filename of Person class”)

Class Pet extends Person

{

}

A computer screen with text on it

Description automatically generated

A computer screen shot of a program code

Description automatically generated

Rules:

1. When parent class has constructor, then child class should implement the same kind of constructor.
2. First statement in child constructor is calling the parent constructor using super()